

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-175844

(43)Date of publication of application : 24.06.1994

(51)Int.Cl.

G06F 9/06

G06F 11/28

(21)Application number : 04-350363

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 03.12.1992

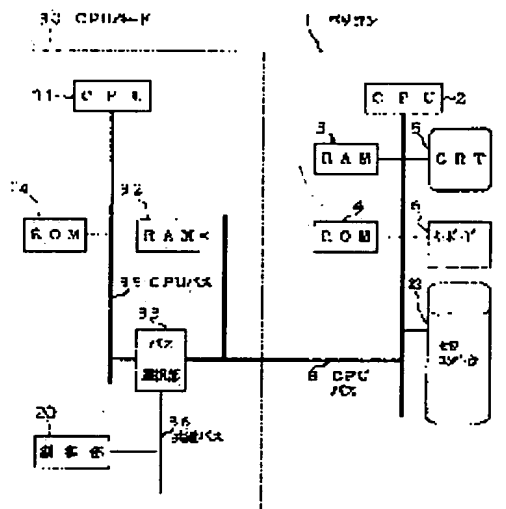
(72)Inventor : FUJII MASAYASU

(54) METHOD FOR DEVELOPING SOFTWARE

(57)Abstract:

PURPOSE: To improve efficiency for the program development of a CPU board by performing program development for the CPU board, control part test and debugging while using a self compiler at a personal computer, and operating the CPU board while transferring a program to the CPU board after debugging on the personal computer is completed.

CONSTITUTION: A CPU bus 8 of a personal computer 1 is connected to a CPU board 30, and a software on the side of the CPU board 30 is developed by a self compiler 22 of the personal computer 1, transferred to the side of the CPU board 30 and installed. Then, the personal computer 1 generates a program for operating a control part 20 with the self compiler 22. Next, the control part 20 is actually operated by loading the program to a RAM 3 and executing it, and the program is debugged. After debugging is completed, the program is transferred to a board side RAM 32 and executed by a board side CPU 31. On the other hand, the control part 20 can be operated even on the side of the board 30 as well.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's]

decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japan Patent Office

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-175844

(43)公開日 平成6年(1994)6月24日

(51)Int.Cl.⁵

G 0 6 F 9/06

11/28

識別記号

4 4 0 S 9367-5B

3 3 0 A 9290-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数3(全14頁)

(21)出願番号 特願平4-350363

(22)出願日 平成4年(1992)12月3日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 藤井 正泰

神戸市兵庫区和田崎町1丁目1番2号 三

菱電機株式会社制御製作所内

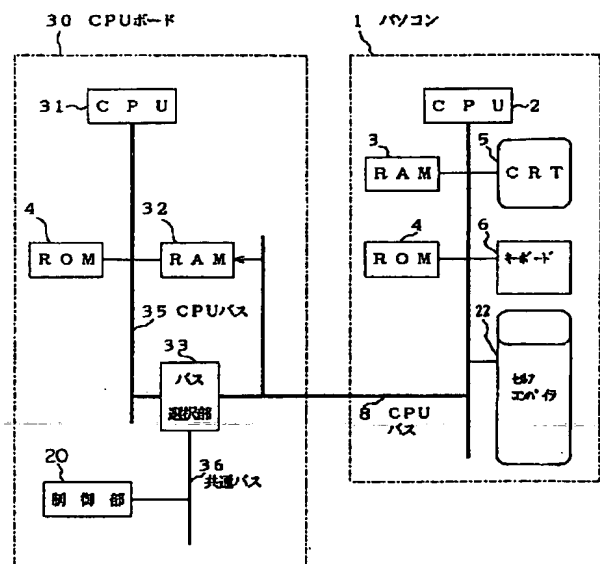
(74)代理人 弁理士 宮園 純一

(54)【発明の名称】 ソフトウェアの開発方法

(57)【要約】

【目的】 試験用のサブCPUボードを必要としないで、ソフトウェア開発ができるようにする。

【構成】 パソコン1のCPUバス8をCPUボード30に接続し、パソコン1のセルフコンパイラ22でCPUボード30側のプログラムのデバッグを行った後CPUボード30側に転送して、CPUボード30を動作可能とする。



【特許請求の範囲】

【請求項1】 CPUを内蔵するCPUボードのソフトウェアの開発において、CPUボードの制御部をパーソナルコンピュータのCPUバスに接続した状態で、パーソナルコンピュータ側のセルフコンパイラで作成した上記制御部プログラムの動作試験及びデバッグをこのパーソナルコンピュータ側で行い、上記CPUボード側のメモリにデバッグ後の制御部プログラムを転送、インストールし、制御部を動作可能としたことを特徴とするソフトウェアの開発方法。

【請求項2】 CPUを内蔵するCPUボードのソフトウェアの開発において、CPUボードの制御部をパーソナルコンピュータのCPUバスに接続した状態で、パーソナルコンピュータ側のセルフコンパイラで作成した上記制御部プログラムの動作試験及びデバッグをこのパーソナルコンピュータ側で行い、上記CPUボード側のメモリにデバッグ後の制御部プログラムを転送、インストールし、制御部を動作可能として、ソフトウェアを開発するとともに、この開発時に上記パーソナルコンピュータのセルフコンパイラにて作成した標準入出力プログラムが、ボード側のCPUにて動作できるよう、標準入出力の疑似回路を設けることで入出力環境をパーソナルコンピュータのCPUに合わせ、入力元をCPUバス延長にて接続したパーソナルコンピュータのキーボードにて、出力先をCPUバス延長にて接続したパーソナルコンピュータのディスプレイにすることを特徴とするソフトウェアの開発方法。

【請求項3】 CPUを内蔵するCPUボードのソフトウェアの開発において、CPUボードの制御部をパーソナルコンピュータのCPUバスに接続した状態で、パーソナルコンピュータ側のセルフコンパイラで作成した上記制御部プログラムの動作試験及びデバッグをこのパーソナルコンピュータ側で行い、上記CPUボード側のメモリにデバッグ後の制御部プログラムを転送、インストールし、制御部を動作可能として、ソフトウェアを開発するとともに、この開発時に上記パーソナルコンピュータのセルフコンパイラにて作成した制御部プログラムを共通バス部におき、制御部からの割込信号をボード側のCPUに入れ、プログラムの割込処理の部分はボード側のCPUが実行し、割込処理以外の通常処理の部分はパーソナルコンピュータ側が実行するようにしたことを特徴とするソフトウェアの開発方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、制御部を有するCPUボードのソフトウェアの開発に関するものである。

【0002】

【従来の技術】 図6は例えば特開昭56-132647号公報に示された従来のソフトウェアの開発方法を説明するためのCPUボードを示す構成図であり、図におい

て、1はパーソナルコンピュータ（以下パソコンと略す）であり、CPU2、RAM3、ROM4、ディスプレイ5、キーボード6、クロスコンパイラ（他のCPUの機械語プログラムをコンパイルして作成するソフトウェア）7、CPUバス8にて構成される。9はサブCPUボードであり、CPU10、CPU制御回路11、バス選択部12、サブ側CPUバス13、パソコン側CPUバス8、共通バス14、接続部15にて構成される。16はテストCPUボードであり、CPU17、RAM18、ROM19、制御部20、CPUバス21にて構成される。

【0003】 次にソフトウェアの開発の手順について説明する。テストCPUボード16から、CPU17を取り外し、代わりに、サブCPUボード9の接続部15をCPUソケットに接続する。パソコン1のクロスコンパイラ7にて作成したテスト側CPU17のプログラムをバス選択部12がパソコン側CPUバス8を選択した状態で、テスト側RAM18に転送する。次に、バス選択部12がサブ側CPUバス13を選択した状態で、サブCPU10に該プログラムを実行させる。サブCPU10動作中において、パソコン1のキーボード6からの命令は、パソコンCPUバス8に接続されたサブCPU制御回路11でサブCPU10に伝えられ、サブCPU10の実行結果はサブCPU制御回路11からパソコンCPUバス8に送られ、パソコン1のディスプレイ5に表示される。

【0004】

【発明が解決しようとする課題】 従来のソフトウェアの開発は以上のようになされているので、テストCPUボードのプログラム開発には、専用のクロスコンパイラが必要で、市販のコンパイラは使えなかった。また、テストCPUボードを試験するためには試験専用のサブCPUボードが必要である等の問題があった。

【0005】 この発明は上記のような課題を解決するためになされたものであり、専用のセルフコンパイラを用いることなく、パソコン用のセルフコンパイラを利用して、ソフトウェア開発ができるようにし、試験用のサブCPUボードを必要としないソフトウェアの開発方法を提供する。

【0006】

【課題を解決するための手段】 この請求項1の発明に係るソフトウェアの開発方法は、CPUボード30を、パソコン1のCPUバス8をCPUボード30に接続し、パソコン1のセルフコンパイラ22でCPUボード30側のソフトウェアの開発を行い、CPUボード30側に転送、インストールする。

【0007】 請求項2の発明では、上記請求項1の発明において、ボード側CPUバス35に標準入出力の疑似回路を設け、入力元をパソコン1のキーボード6、出力先をパソコン1のディスプレイ5にしたものである。

【0008】請求項3の発明では、上記請求項1の発明において、CPUボード30側の制御部20からの割込信号45をボード側CPU31に入れ、制御部プログラムは共通バス36におき、割込処理44はボード側で、割込以外の通常処理43はパソコン1側で実行するようにしたものである。

【0009】

【作用】請求項1の発明では、パーソナルコンピュータのセルフコンパイラを用いてコンパイルし、オブジェクトプログラムの試験、デバッグ完了後、CPUボードのメモリにこのデバッグ後の制御部プログラムを転送して、インストールとし、制御部を動作可能とする。

【0010】請求項2の発明では、セルフコンパイラでコンパイルして作成された制御部プログラムが、CPUボードで動作する場合、標準入出力は、パソコンのディスプレイとキーボードとなる。

【0011】請求項3の発明では、パソコン側の通常処理は、ボード側割込処理によりリアルタイムに変化するワークエリアを参照して行う。

【0012】

【実施例】

実施例1. 以下、この発明の実施例1を図に基づいて説明する。図1において、22はパソコン1に実装されたセルフコンパイラである。30はCPUボードであり、31のCPU、32のRAM、4のROM、33のバス選択部、20の制御部、35のボード側CPUバス、8のパソコン側CPUバス、36の共通バスで構成される。上記制御部20は、多目的用のインターフェースで、汎用I/O、DI/DO、シリアルIO（通信用モジュール）がある。またセルフコンパイラ22は市販品のもので、後述のprintf関数、scanf関数以外に標準入出力用関数、ioctl関数が用意されている。基本的にioctlはメモリI/Oに対してREAD、メモリI/Oに対してWRITEを行うだけなので、制御部20はDI/Oインターフェース、2ポートメモリインターフェース等にする必要がある。

【0013】次に動作について説明する。パソコン1にて、セルフコンパイラ22で制御部20を動作させるプログラムを作成する。次に、RAM3にロードし実行させることで制御部20を実際に動作させ、プログラムのデバッグを行う。デバッグ終了後、ボード側RAM32にプログラムを転送し、ボード側CPU31に実行させる。ボード側、パソコン側とで、制御部20に対するプログラム環境はソフトウェア的にもハードウェア的にも同じなので、ボード側においても制御部20を動作させることができる。すなわち、プログラム製造はつぎの手順イ〜ホのとおりである。

イ、ソースプログラムの製造、つまり高級言語によるプログラムをアスキーコードでファイル化したものの製造を行う。この製造は、各OS用に用意された、テキスト

エディタにて作成キーボード6、CRT5が接続されているパソコン側で作成する。上記エディタにはMS-DOSではMIFES、UNIXではViエディタが有名である。

ロ、イでのプログラム完成後、セルフコンパイラ22を用いてこのソフトをコンパイルして、オブジェクトプログラムを製造する。

ハ、このオブジェクトプログラムをRAM3に記憶する。

ニ、パソコン1のCPU2で、RAM3をアクセスしCRT5、キーボード6を用いて制御部20駆動のための試験及びデバッグ行う。

ホ、デバッグ完了後、RAM3中のデータをCPU8を介してRAM32に転送、インストールする。

この場合、CPU31を用いてのデバッグはデバッグ済みなので行わなくても良い。バス選択部33の切替えは、ハードウェアスイッチを用いて行う。以上のようにセルフコンパイラにて、ソースプログラムをオブジェクトプログラムに変換してパソコン用アプリケーションを作成し、このアプリケーション機械語をRAM3からRAM32に転送しインストールしてCPU31に実行させる。ここで、市販のコンパイラは、パソコンのOSシステムコールを繰り返し行うことで機能を実現する機械語（オブジェクトプログラム）を作成するものである。一般的に、CPUボード作る場合は、OSのシステムコールはボード専用に使っていた。したがって、パソコン用コンパイラの作成する機械語のOSシステムコールを受け付けない。なお、CPU2が文字をCRTに表示する場合、セルフコンパイラの作成する機械語はOSのシステムコールを呼ぶだけである。但し、環境はこのOSのシステムコールのコーリングシーケンス（AHが02H内部割込ベクタ番号が21Hであるということ）が完全に同じでなければならない。

【0014】実施例2. 以下、この発明の実施例2を図2に基づいて説明する。但し、図2中、図1と同じものは同一符号を用いている。図2において、30はCPUボードであり、31のCPU、32のRAM、4のROM、35のボード側CPUバス、8のパソコン側CPUバス、40の標準出力疑似回路、41の標準入力疑似回路で構成される。標準出力疑似回路40、標準入力疑似回路41は、DO、DIとして機能するもので、具体例は図3に示すとおり周知の回路である。同図から明らかなように制御は、CRTコントローラ70、キーボードコントローラ71を介して行われる。なお、72はCRTバス、73はキーボードバスである。このようにディスプレイ5とキーボード6との間にCRTコントローラ70、キーボードコントローラ71が有り、これ等が標準出力疑似回路40、標準入力疑似回路41で制御される。

【0015】次に動作について説明する。実施例と同様

5

にパソコン1のセルフコンパイラ22にて作成したデバッグ完了後のプログラムをボード側RAM32に転送し、インストールして、ボード側CPU31に実行させる。プログラムの標準出力関数（C言語のprintf関数相当）を実行した場合、ボード側CPU31は標準出力疑似回路40に結果を出力し、パソコン側CPU2は、該出力結果をパソコン側CPUバス8経由、ディスプレイ5に表示出力する。また、パソコン側CPU2は、キーボード6の入力を標準入力疑似回路41に常時書き込んでいる。プログラムの標準入力関数（C言語のscanf関数相当）を実行した場合、ボード側CPU31は標準入力疑似回路41をスキャンし取り込む。従って、CPUボード30で動作中に、パソコン1のディスプレイ5と、キーボード6を使用して動作確認ができる。すなわち、標準入力疑似回路41により、キーボード6を操作したのと等価な規定のデータをCPU31に供給してCPU31を駆動させ、標準出力疑似回路40により結果をバス8を介してディスプレイ5に表示させることで、動作を検証可能として、デバッグを容易にする。標準出力疑似回路40、標準入力疑似回路41を用いれば、キーボード6のパラメータを押すだけで、デバッグが可能である。

【0016】実施例3。以下、この発明の実施例3を図に基づいて説明する。図4において、30はCPUボードであり、CPU31、RAM32、ROM4、ボード側CPUバス35、パソコン側CPUバス8、標準出力疑似回路40、標準入力疑似回路41、バス選択部33、制御部20、ボード側CPUバス35、パソコン側CPUバス8、共通バス36、RAM42、通常処理43、割込処理44で構成される。45は制御部20からボード側CPU31に入力される割込信号である。

【0017】次に動作について説明する。パソコン1のセルフコンパイラ22にて作成した制御部20のプログラムをボード側RAM42にプログラムを転送し、パソコン側CPU2で制御部20の試験を行う。制御部プログラムには、通常処理43と割込処理44がリンクされ1つのロードモジュールとなっている。制御部20からの割込信号45がボード側CPU31に入っているため、割込処理部44はボード側CPU31によって実行される。通常処理43はパソコン側CPU2が実行する。通常処理43と割込処理44はセルフコンパイラ22によりリンクされているため、通常処理43は容易に割込処理44内ワークエリアを参照することができる。このように、割込処理はCPUボード側で動作させ、パソコンでの制御部試験と組合せることで試験効率をあげることができる。このように、本実施例ではCPU31で割込処理44を実行させ、CPU2で通常処理43を実行させて、分担を図るものであるが、通常処理（割込処理以外の処理）43と割込処理44とを明確に分け

6

て、記憶するための具体的方法は、図5に示す如く周知の割込ベクタに登録することにより行われる。これにより、通常処理では割込処理をCALLしない。割込処理は割込ベクタにアドレス登録され、割込信号がCPU31に入った時、CPU31がジャンプ実行する。

【0018】

【発明の効果】以上のように、請求項1の発明によれば、パソコンでセルフコンパイラを使用してCPUボード用のプログラム開発、および制御部試験、デバッグを行い、パソコン上でのデバッグ終了後、CPUボードにプログラムを転送し動作させるようにしたので、CPUボードのプログラム開発を効率化できる。

【0019】請求項2の発明によれば、標準入出力回路を設けて標準入出力プログラムがボード側のCPUで動作できるようにしたので、CPUボードで動作中に、パソコンのディスプレイとキーボードを使用して動作確認ができ、動作確認を容易に行える。

【0020】請求項3の発明によれば、割込処理は、CPUボード側で動作させ、通常処理はパーソナルコンピュータ側で行なえるようにしたので、パソコンでの制御部試験と組み合わせることができ、試験効率をさらに上げることができる。

【図面の簡単な説明】

【図1】この発明の実施例1を示すシステム構成図である。

【図2】この発明の実施例2を示すシステム構成図である。

【図3】第2の実施例の具体例を示す回路図である。

【図4】この発明の実施例3を示すシステム構成図である。

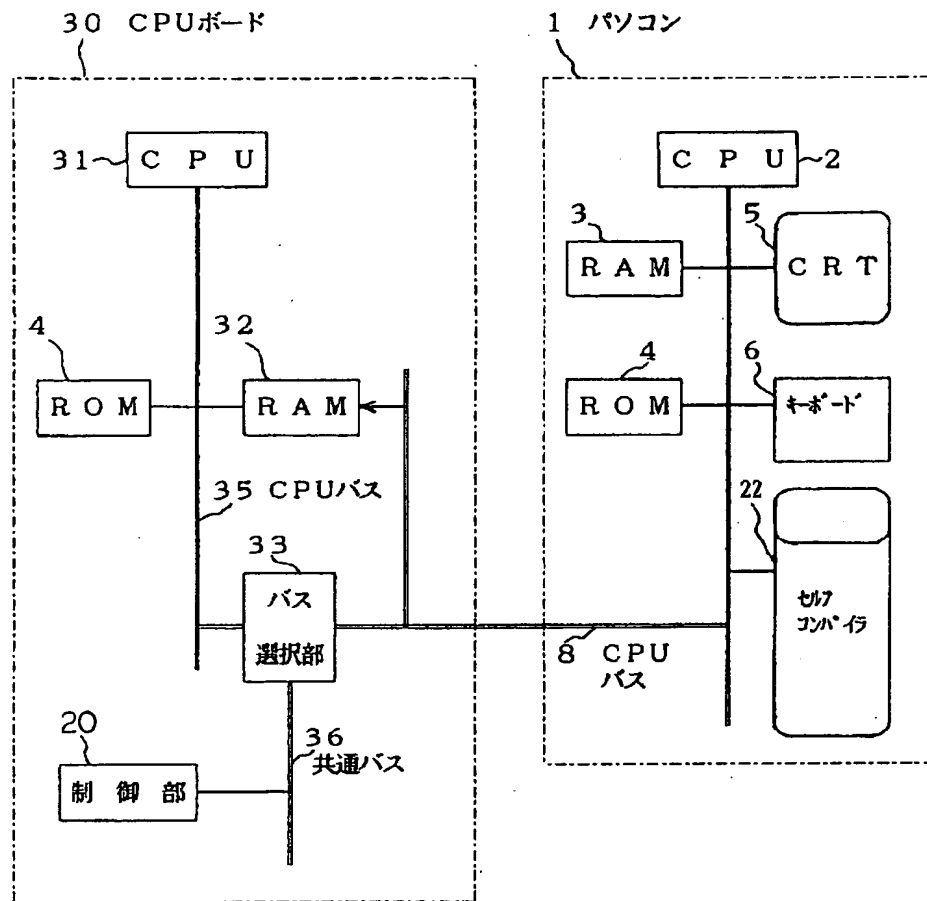
【図5】第3の実施例の具体例を示す説明図である。

【図6】従来のソフトウェアの開発方法の一例を示すシステム構成図である。

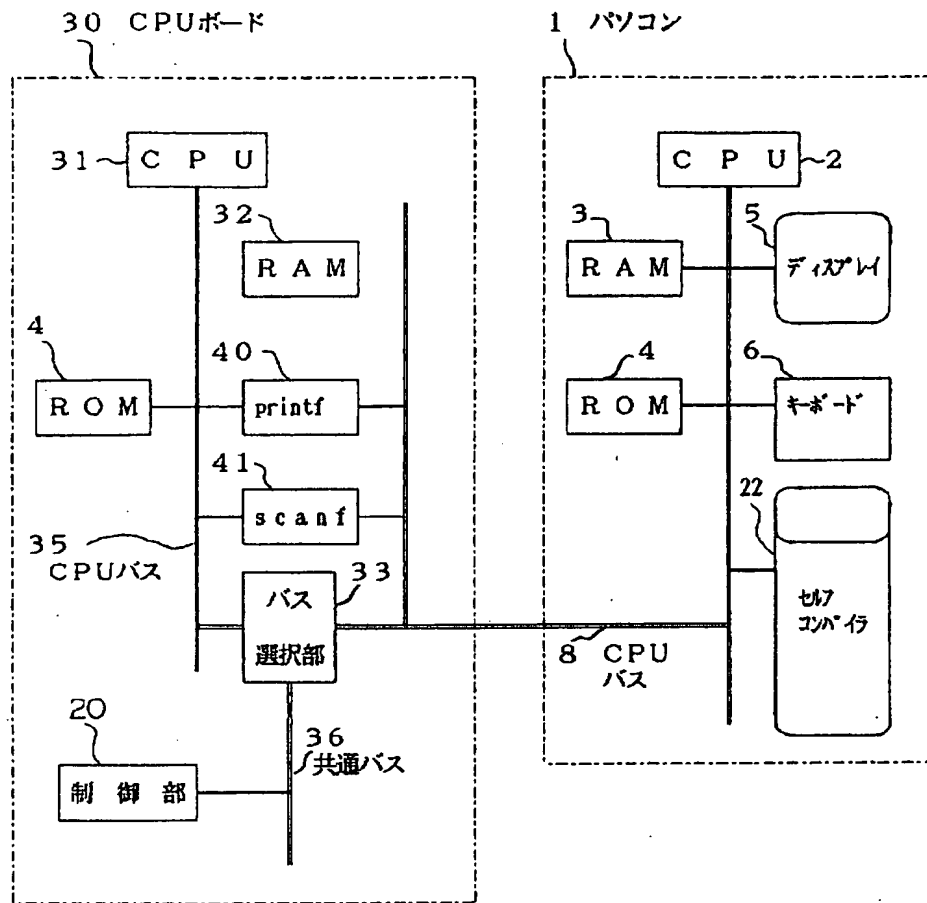
【符号の説明】

1 パソコン
5 ディスプレイ
6 キーボード
8 パソコン側CPUバス
20 制御部
22 セルフコンパイラ
30 CPUボード
31 ボード側CPU
35 ボード側CPUバス
36 共通バス
40 標準出力疑似回路
41 標準入力疑似回路
43 通常処理
44 割込処理
45 割込信号

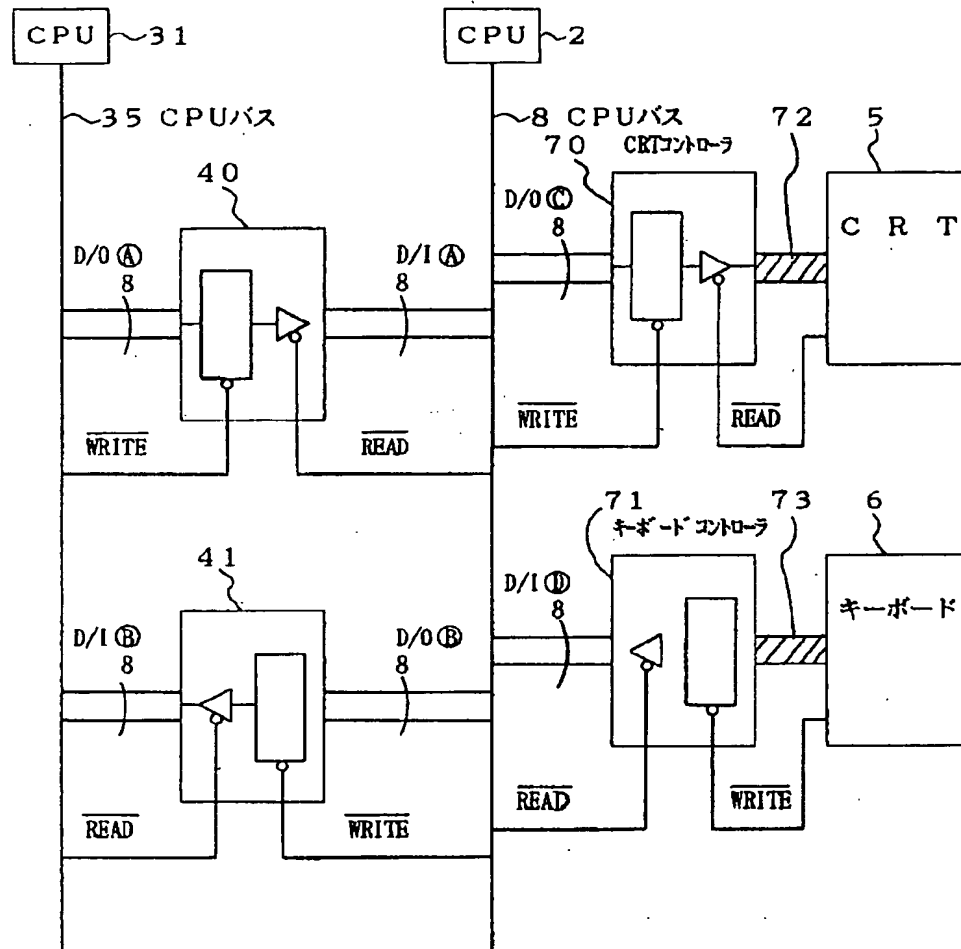
【図1】



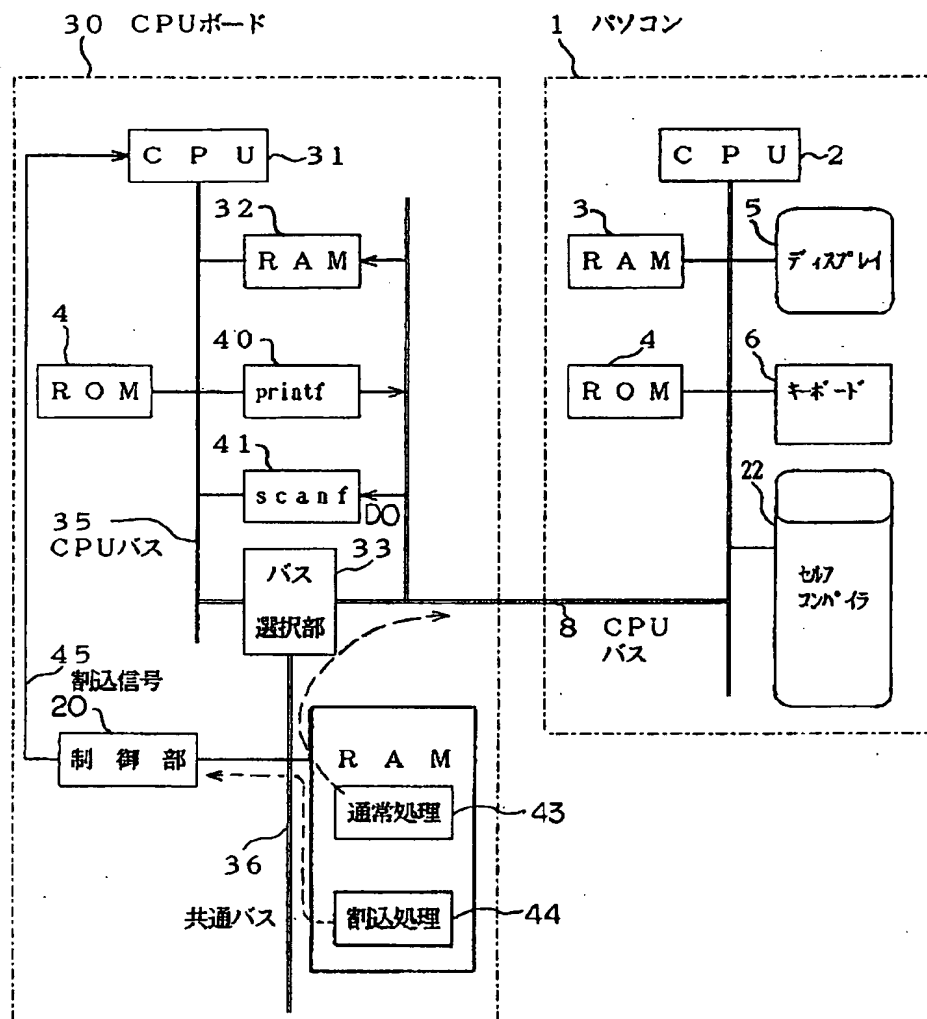
【図2】



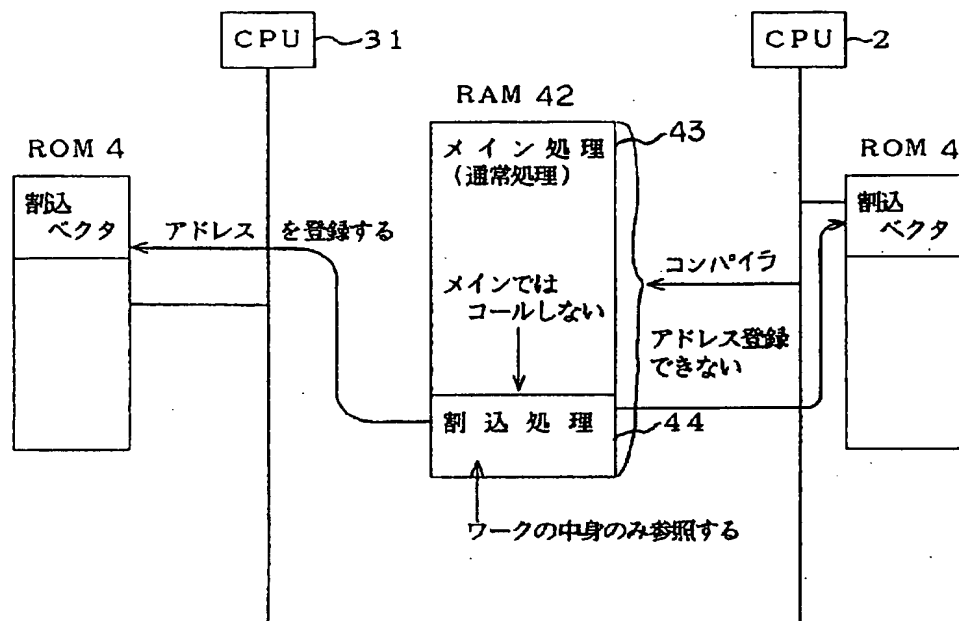
【図3】



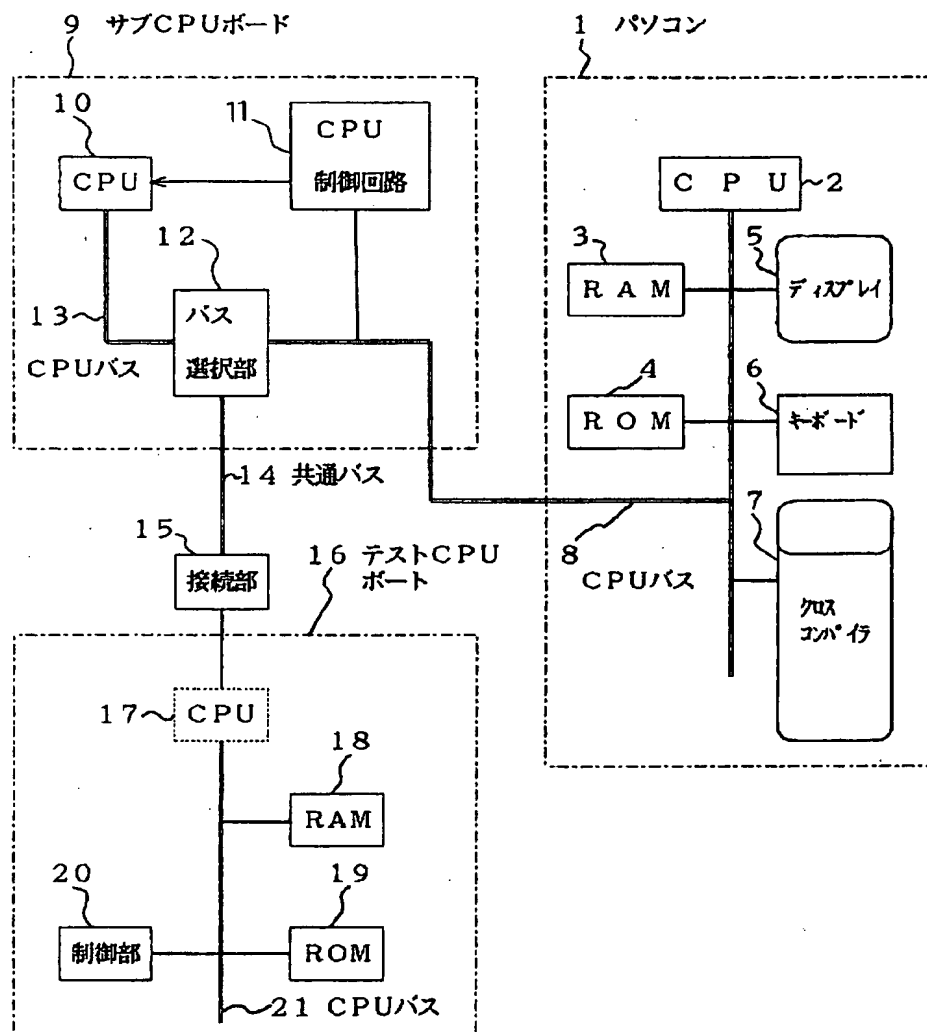
【図4】



【図 5】



【図6】



【手続補正書】

【提出日】平成5年5月20日

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】0005

【補正方法】変更

【補正内容】

【0005】この発明は上記のような課題を解決するためになされたものであり、専用のクロスコンパイラを用いることなく、パソコン用のセルフコンパイラを利用して、ソフトウェア開発ができるようにし、試験用のサブCPUボードを必要としないソフトウェアの開発方法を提供する。

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0006

【補正方法】変更

【補正内容】

【0006】

【課題を解決するための手段】この請求項1の発明に係るソフトウェアの開発方法は、パソコン1のCPUバス8をCPUボード30に接続し、パソコン1のセルフコンパイラ22でCPUボード30側のソフトウェアの開発を行い、CPUボード30側に転送、インストールする。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0013

【補正方法】変更

【補正内容】

【0013】次に動作について説明する。パソコン1にて、セルフコンパイラ22で制御部20を動作させるプログラムを作成する。次に、RAM3にロードし実行させることで制御部20を実際に動作させ、プログラムのデバッグを行う。デバッグ終了後、ボード側RAM32にプログラムを転送し、ボード側CPU31に実行させる。ボード側、パソコン側とで、制御部20に対するプログラム環境はソフトウェア的にもハードウェア的にも同じなので、ボード側においても制御部20を動作させることができる。すなわち、プログラム製造はつぎの手順イ〜ホのとおりである。

イ、ソースプログラムの製造、つまり高級言語によるプログラムをアスキーコードでファイル化したものの製造を行う。この製造は、各OS用に用意された、テキストエディタにて作成キーボード6、CRT5が接続されているパソコン側で作成する。上記エディタにはMS-DOSではMIFES、UNIXではViエディタが有名である。

ロ、イでのプログラム完成後、セルフコンパイラ22を用いてこのソフトをコンパイルして、オブジェクトプログラムを製造する。

ハ、このオブジェクトプログラムをRAM3に記憶する。

ニ、パソコン1のCPU2で、RAM3をアクセスしCRT5、キーボード6を用いて制御部20駆動のための

試験及びデバッグ行う。

ホ、デバッグ完了後、RAM3中のデータをCPU8を介してRAM32に転送、インストールする。

この場合、CPU31を用いてのデバッグはデバック済みなので行わなくても良い。バス選択部33の切替えは、ハードウェアスイッチを用いて行う。以上のようにセルフコンパイラにて、ソースプログラムをオブジェクトプログラムに変換してパソコン用アプリケーションを作成し、このアプリケーション機械語をRAM3からRAM32に転送しインストールしてCPU31に実行させる。ここで、市販のコンパイラは、パソコンのOSシステムコールを繰り返し行うことで機能を実現する機械語（オブジェクトプログラム）を作成するものである。一般的に、CPUボード作る場合は、OSのシステムコールはボード専用に作っていた。したがって、パソコン用コンパイラの作成する機械語のOSシステムコールを受け付けない。なお、CPU2が文字をCRTに表示する場合、セルフコンパイラの作成する機械語はOSのシステムコールを呼ぶだけである。但し、環境はこのOSのシステムコールのコーリングシーケンス（OSがMS-DOSであるならば、AHが02H、内部割込ベクタ番号が21Hであるということ）が完全に同じでなければならない。

【手続補正4】

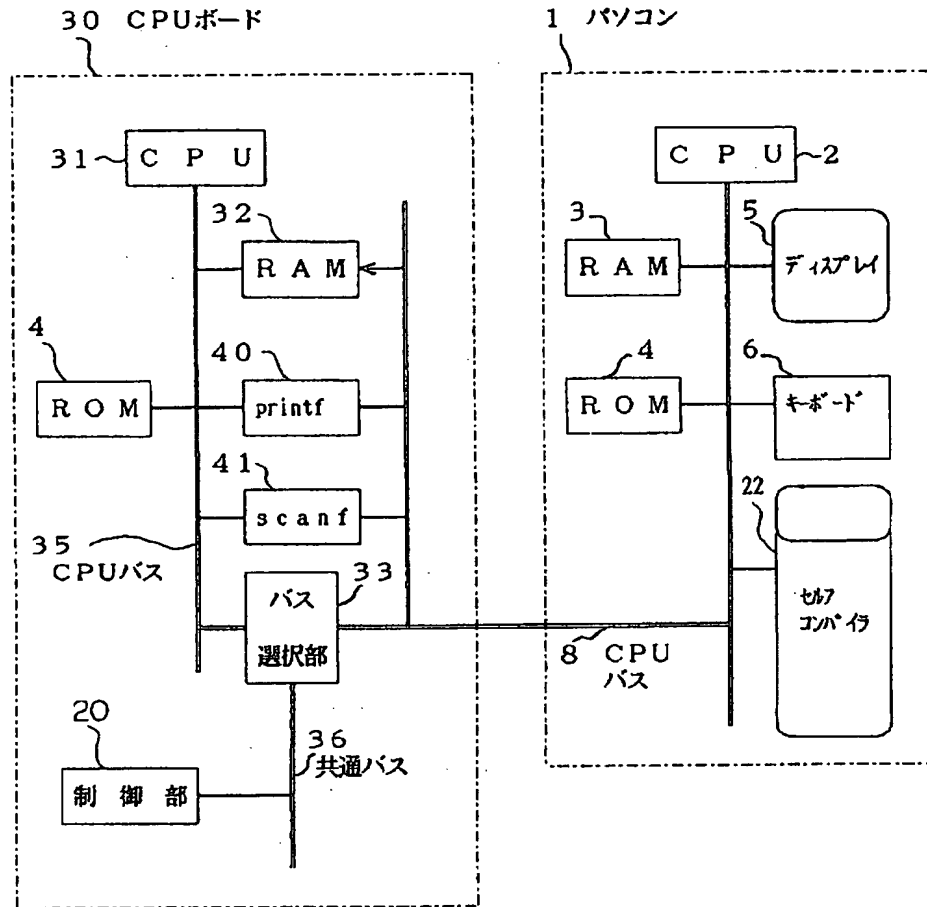
【補正対象書類名】図面

【補正対象項目名】図2

【補正方法】変更

【補正内容】

【図2】



【手続補正5】

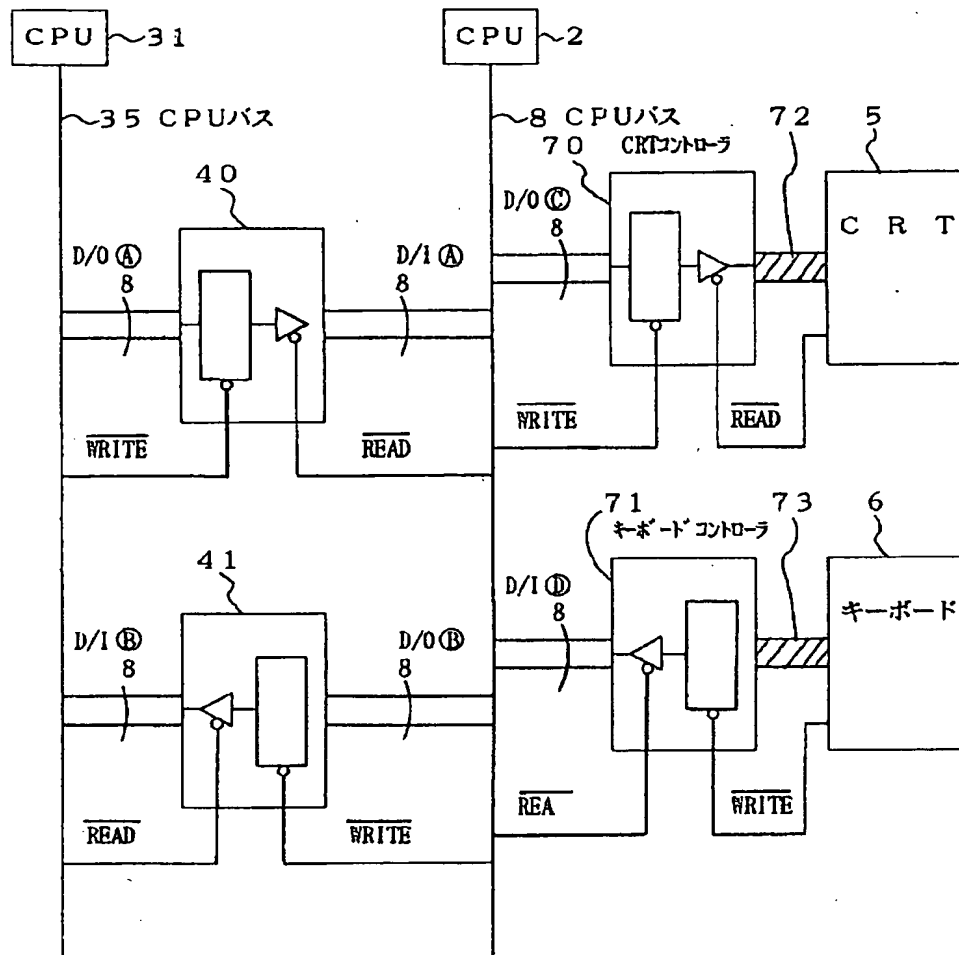
【補正対象書類名】図面

【補正対象項目名】図3

【補正方法】変更

【補正内容】

【図3】



【手続補正6】

【補正対象書類名】図面

【補正対象項目名】図4

【補正方法】変更

【補正内容】

【図4】

